Fairness via In-Processing in the Over-parameterized Regime: A Cautionary Tale

Anonymous authors
Paper under double-blind review

Abstract

The success of deep learning is driven by the counter-intuitive ability of over-parameterized deep neural networks (DNNs) to generalize, even when they have sufficiently many parameters to perfectly fit the training data. In practice, test error often continues to decrease with increasing over-parameterization, a phenomenon referred to as double descent. This allows deep learning engineers to instantiate large models without having to worry about over-fitting. Despite its benefits, however, prior work has shown that over-parameterization can exacerbate bias against minority subgroups. Several fairness-constrained DNN training methods have been proposed to address this concern. Here, we critically examine MinDiff, a fairness-constrained training procedure implemented within TensorFlow's Responsible AI Toolkit, that aims to achieve Equality of Opportunity. We show that although MinDiff improves fairness for under-parameterized models, it is likely to be ineffective in the over-parameterized regime. This is because an overfit model with zero training loss is trivially group-wise fair on training data, creating an "illusion of fairness," thus turning off the MinDiff optimization (this will apply to any disparity-based measures which care about errors or accuracy. It won't apply to demographic parity). We find that within specified fairness constraints, under-parameterized MinDiff models can even have lower error compared to their over-parameterized counterparts (despite baseline over-parameterized models having lower error compared to their under-parameterized counterparts). We further show that MinDiff optimization is very sensitive to choice of batch size in the under-parameterized regime. Thus, fair model training using MinDiff requires time-consuming hyper-parameter searches. Finally, we suggest using previously proposed regularization techniques, viz. L2, early stopping and flooding in conjunction with MinDiff to train fair over-parameterized models. In our results, over-parameterized models trained using MinDiff+regularization with standard batch sizes are fairer than their under-parameterized counterparts, suggesting that at the very least, regularizers should be integrated into fair deep learning flows.

1 Introduction

Over the past few years, machine learning (ML) solutions have found wide applicability in wide range of domains. However, recent work has shown that ML methods can exhibit unintended biases towards specific population groups, for instance in applications like hiring (Schumann et al., 2020), credit verification (Khandani et al., 2010), facial recognition (Buolamwini & Gebru, 2018; Grother et al., 2010; Ngan & Grother, 2015), recidivism prediction (Chouldechova, 2017) and recommendation systems (Biega et al., 2018; Singh & Joachims, 2018), resulting in negative societal consequences. To address this concern, there is an growing and influential body of work on mitigating algorithmic unfairness of ML models. These solutions are being integrated within widely used ML frameworks and are beginning to find practical deployment (AI; Akihiko Fukuchi, 2020). As ML fairness methods make the transition from theory to practice, their ability to achieve stated goals in real-world deployments merits closer examination.

Methods to train fair models can be broadly categorized based on the stage at which they are deployed: pre-training, in-training, or post-training. Of these, only in-training methods substantively modify the model training process. This paper examines the performance of MinDiff (Prost et al., 2019), the principal

in-training method integrated within TensorFlow's Responsible AI Framework (AI). We are particularly interested in MinDiff for training fair deep learning models because, given TensorFlow's widespread adoption, there is good reason to believe that it will be picked as the default choice by practitioners working within this framework.

We evaluate MinDiff on two datasets, Waterbirds and CelebA that are commonly used in fairness literature, and observe several notes of caution. Because the success of deep learning can be attributed at least in part to the surprising ability of over-parameterized deep networks (networks with sufficiently many parameters to *memorize* the training dataset) to generalize (Nakkiran et al., 2020), we begin by evaluating the relationship between model capacity and fairness with MinDiff. We observe that MinDiff does increase fairness for small, under-parameterized models, but is almost entirely ineffective on larger over-parameterized networks. Thus, in some cases, under-parameterized MinDiff models can have lower fairness-constrained error compared to their over-parameterized counterparts even though over-parameterized models are always better on baseline error (i.e., error on models trained without MinDiff optimization). We caution that when using MinDiff for fairness, ML practitioners must carefully choose model capacity, something which is generally unnecessary when fairness is not a concern and the goal is simply to minimize error.

We find the reason MinDiff is ineffective in the over-parameterized regime is because an overfit model with zero training loss means any disparity-based unfairness necessarily goes to zero on the training dataset, creating an "illusion of fairness" during training, thus turning off the MinDiff optimization (this will apply to any disparity-based measures which care about errors or accuracy. It won't apply to demographic parity). Thus, we explore whether strong regularization used along with MinDiff can alleviate its ineffectiveness. Specifically, we consider two classes of regularization techniques: implicit (batch sizing (Smith et al., 2021; Barrett & Dherin, 2021) and early stopping (Morgan & Bourlard, 1990)) and explicit (weight decay (Krogh & Hertz, 1992) and a recently proposed "loss flooding" method (Ishida et al., 2020)) regularizers. We find that: (1) batch sizing only helps for medium sized models around the interpolation threshold; (2) the remaining three methods all improve fairness in the over-parameterized regime; (3) early-stopping and flooding result in the fairest models for the Waterbirds and CelebA datasets, respectively; and (4) with effective regularization, over-parameterized models are fairer than their under-parameterized counterparts.

2 Related Work

There are several techniques in literature to mitigate algorithmic bias. These techniques can be broadly categorized as: pre-processing, in-processing and post-processing. Pre-processing techniques aim to de-identify sensitive information and create more balanced training datasets (Quadrianto et al., 2019; Ryu et al., 2018; Feldman et al., 2015; Wang & Deng, 2019; Karkkainen & Joo, 2021; Dixon et al., 2018). In-processing (Prost et al., 2019; Cherepanova et al., 2021; Sagawa et al., 2020a;b; Padala & Gujar, 2021; Agarwal et al., 2018; Zafar et al., 2019; Donini et al., 2018; Lahoti et al., 2020; Beutel et al., 2019; Martinez et al., 2020; Wadsworth et al., 2018; Goel et al., 2018; Wang & Deng, 2019; Hashimoto et al., 2018) techniques alter the training mechanism by imposing fairness constraints to the training objective, or utilize adversarial training (Beutel et al., 2017; Zhang et al., 2018; Madras et al., 2018) to make predictions independent of sensitive attributes. Post-processing techniques (Hardt et al., 2016b; Wang et al., 2020; Savani et al., 2020; Chzhen et al., 2019; Jiang et al., 2020; Wei et al., 2020) alter the outputs of an existing model, for instance, using threshold correction (Zhou & Liu, 2006; Collell et al., 2016; Menon et al., 2021a) that applies different classification thresholds to each sensitive group (Hardt et al., 2016b). In this paper, we focus on MinDiff (Prost et al., 2019), the primary in-processing procedure implemented within TensorFlow's Responsible AI toolkit. While our quantitative conclusions might differ, we believe that similar qualitative conclusions will hold for other in-processing methods because overfit models are trivially fair.

With the growing adoption of large over-parameterized deep networks, recent efforts have sought to investigate their fairness properties (Menon et al., 2021b; Sagawa et al., 2020a; Cherepanova et al., 2021; Sagawa et al., 2020b). Pham et al. (2021) observed that over-parameterized ERM models have better worst-group generalization compared to their under-parameterized counterparts. However, Maity et al. (2022) warn that baseline ERM models should not be considered state-of-the-art to train fair over-parameterized models. Sagawa et al. (2020b) proposed a pre-processing technique by investigating the role of training data

characteristics (such as ratio of majority to minority groups and relative informativeness of spurious versus core features) on fairness and observed that sub-sampling improves fairness in the over-parameterized regime. Menon et al. (2021b) found that post-processing techniques including retraining with sub-sampled majority groups and threshold correction also enhance fairness in over-parameterized models. Cherepanova et al. (2021) report that in-processing convex surrogates of fairness constraints like equal loss, equalized odds penalty, disparate impact penalty, etc., (Padala & Gujar, 2021) are ineffective on over-parameterized models, but do not propose any techniques to increase the effectiveness of in-processing methods. Wald et al. (2022) theoretically show that interpolating models cannot satisfy fairness constraints. However, it is not thoroughly investigated how fairness constraints can be effectively implemented in over-parameterized models. It is possible that using methods such as MinDiff may improve the training of fair over-parameterized models. Our work is the first to systematically compare under- vs. over-parameterized deep models trained using in-processing fairness methods using MinDiff as a representative method.

Regularization techniques (Morgan & Bourlard, 1990; Srivastava et al., 2014; Krogh & Hertz, 1992; Ishida et al., 2020) are popularly used in deep learning frameworks to avoid over-fitting. Lately, researchers have also exploited the benefits of regularizers to train fair models. For example, Sagawa et al. (2020a) proposed distributionally robust optimization (DRO) to improve worst-group generalization, but they observed that their approach fails if training loss converges to zero. Hence, they use L2 weight regularization and early stopping to improve fairness in the over-parameterized regime. Our paper systematically evaluates different regularizers, including batch sizing, early stopping, weight decay and the recently proposed flooding loss (Ishida et al., 2020) for MinDiff training across different model sizes, and makes several new observations about the role of regularization in enhancing fairness.

3 Methodology

We now describe our evaluation methodology.

3.1 Setup

In this paper, we consider binary classification problem on a training dataset $\mathcal{D} = \{x_i, a_i, y_i\}_{i=1}^{N}$, where x_i is an input (an image for instance) $a_i \in \{0, 1\}$ is a sensitive attribute of the input, and $y_i \in \{0, 1\}$ is the corresponding ground-truth label. The training data is sampled from a joint distribution $P_{X,A,Y}$, over random variables X, A, and Y. Deep neural network (DNN) classifiers are represented as a parameterized function $f_\theta : \mathcal{X} \to [0, 1]$, where θ are trainable parameters, obtained in practice by minimizing the binary cross-entropy loss function $\mathcal{L}_P$:

$$\mathcal{L}_P = -\frac{1}{N}\sum_{i=1}^{N}[y_i \cdot log(f_\theta(x_i)) + (1 - y_i) \cdot log(1 - f_\theta(x_i))] \tag{1}$$

via stochastic gradient descent.

We denote the classification threshold as τ which can be used to make predictions $\hat{f}_\theta(x; \tau)$ as shown below

$$\hat{f}_\theta(x; \tau) = \begin{cases} 1, & f_\theta(x) \geq \tau \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Standard DNN training methods seek to achieve low test error $\mathbb{P}[\hat{f}_\theta(X; \tau) \neq Y]$ (typically, $\tau = 0.5$) but performance conditioned on sensitive attributes can vary, leading to outcomes that are biased in favor of or against specific sub-groups. Several fairness metrics have been defined in prior work to account for this bias; in this paper, we will use the widely adopted equality of opportunity metric (Hardt et al., 2016a).

Equality of Opportunity (Hardt et al., 2016a) is a widely adopted fairness notion that seeks to equalize false negative rates (FNR) across sensitive groups. For binary sensitive attributes the FNR_{gap} is defined as:

$$\text{FNR}_{\text{gap}} = \big|\mathbb{P}[\hat{f}_\theta(X; \tau) = 0|Y = 1, A = 0] - \mathbb{P}[\hat{f}_\theta(X; \tau) = 0|Y = 1, A = 1]\big|. \tag{3}$$

As we describe next, MinDiff (and several other methods) seek to minimize the FNR_{gap} during training.

3.2 MinDiff Training

MinDiff (Prost et al., 2019) is an in-processing optimization framework that seeks to achieve a balance between two objectives: low test error and low FNR_{gap}. For this, MinDiff proposes a modified loss function $\mathcal{L}_T = \mathcal{L}_P + \lambda \mathcal{L}_M$, where $\mathcal{L}_T$ is the total loss, that is a weighted sum of the cross-entropy loss, defined in Equation 1, and $\mathcal{L}_M$, a differentiable proxy for the FNR_{gap}. In the modified loss function, $\lambda \in \mathbb{R}_+$ is a user-defined parameter that controls the relative importance of the fairness versus test error. The fairness term in the modified loss function, $\mathcal{L}_M$, uses the maximum mean discrepancy (MMD) distance between the neural network's outputs for the two sensitive groups when $Y = 1$, i.e.,

$$\mathcal{L}_M = \text{MMD}(f_\theta(X)|A = 0, Y = 1, f_\theta(X)|A = 1, Y = 1). \tag{4}$$

We refer the reader to the original MinDiff paper for a formal definition of the MMD distance (Prost et al., 2019).

3.3 Post-hoc Threshold Correction

Due to fairness requirements of the application, ML practitioners might seek to train models with an FNR_{gap} lower than a specified threshold Δ_{FNR}. However, MinDiff does not explicitly constrain the FNR_{gap}; this can be addressed using an additional post-processing threshold correction step, as proposed by (Hardt et al., 2016b). The idea is to use different classification thresholds for each sub-group, $\tau_{A=0}$ and $\tau_{A=1}$, that are selected such that the FNR_{gap} constraint is met *and* test error is minimized.:

$$\left|\mathbb{P}[\hat{f}_\theta(X; \tau_{A=0}) = 0|Y = 1, A = 0] - \mathbb{P}[\hat{f}_\theta(X; \tau_{A=1}) = 0|Y = 1, A = 1]\right| \leq \Delta_{\text{FNR}}, \tag{5}$$

The two thresholds can be picked using grid search on a validation dataset; we refer to the resulting test error as the **fairness-constrained test error**. Pareto front of fairness-constrained test error and Δ_{FNR} is used to compare different model sizes and fairness methods. In particular, a model or method is fairer if it has lower fairness-constrained test error compared to the alternative for a fixed Δ_{FNR}.

3.4 Regularization Techniques

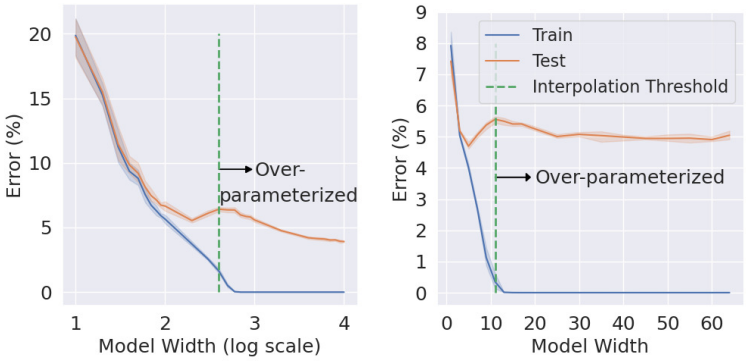
We evaluate four regularization techniques to improve performance of MinDiff on over-parameterized networks.

Reduced Batch Sizes: Due to the stochastic nature of SGD, smaller batch sizes can act as implicit regularizers during training (Smith et al., 2021; Barrett & Dherin, 2021). This is because smaller batch sizes provide a noisy estimate of the total loss $\mathcal{L}_T$.

Weight Decay: Weight decay (Krogh & Hertz, 1992) explicitly penalizes the parameters θ of the DNN from growing too large. Weight decay adds a penalty, usually the L2 norm of the weights, to the loss function.

Early Stopping: Early stopping (Morgan & Bourlard, 1990) terminates DNN training earlier than the point at which training loss converges to a local minima, and has been shown to be particularly effective for over-parameterized deep networks (Li et al., 2019). A common implementation of early stopping is to terminate training once the validation loss has increased for a certain number of gradient steps or epochs (Morgan & Bourlard, 1990). For models trained with MinDiff, we explore two versions of early stopping in which we use either the primary loss, $\mathcal{L}_P$, or total loss, $\mathcal{L}_T$, as a stopping criterion.

Flooding Regularizer: Finally, motivated by our goal to prevent the primary training loss from going to zero (which then turns off MinDiff as well), we apply the flooding regularizer (Ishida et al., 2020) that encodes this goal *explicitly*. Flooding operates by performing gradient descent only if $\mathcal{L}_P > b$, where b is the flood level. Otherwise, if $\mathcal{L}_P \leq b$, then gradient ascent takes place as shown in Equation 6. This phenomenon ensures



(a) Waterbirds (b) CelebA

Figure 1: We show the model-wise double descent behaviour on baseline models trained using (a) Waterbirds and (b) CelebA datasets respectively. Interpolation threshold (shown in green dotted line) is the point where the model is large enough to fit the training data. The region beyond the interpolation threshold is called the over-parameterized regime.

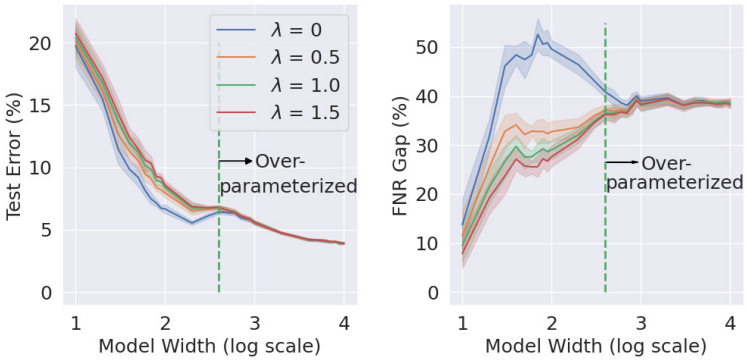


Figure 2: We show the (a) average test error, and (b) FNR_{gap} versus model width for MinDiff optimization with $\lambda = \{0.0, 0.5, 1.0, 1.5\}$'s on Waterbirds dataset. MinDiff optimization has negligible to no impact on fairness in the over-parameterized models. However, for under-parameterized models, we find that increasing λ substantially reduces the FNR_{gap}.

that $\mathcal{L}_P$ floats around the flood level b and never approaches zero. We implement flooding by replacing the primary loss term in $\mathcal{L}_T$ with a new loss term:

$$\mathcal{L}'_P = \left|\mathcal{L}_P - b\right| + b, \tag{6}$$

which, in turn, enables continued minimization of the MinDiff loss term, $\mathcal{L}_M$, over the training process.

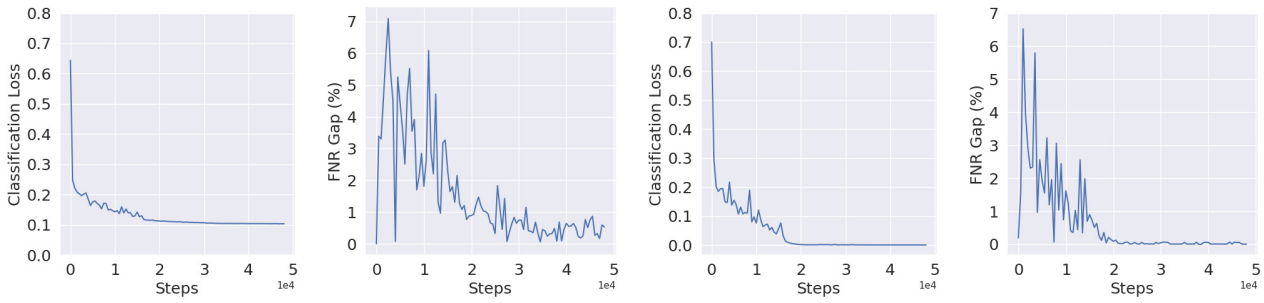
4 Experimental Setup

We perform our experiments on the Waterbirds (Sagawa et al., 2020a) and CelebA (Liu et al., 2015) datasets which have previously been used in fairness evaluations of deep learning models. Here, we describe network architectures, training and evaluations for the two datasets.

4.1 Waterbirds Dataset

Waterbirds is synthetically created dataset (Sagawa et al., 2020a) which contains water- and land-bird images overlaid on water and land backgrounds. A majority of waterbirds (landbirds) appear in water (land) backgrounds, but in a minority of cases waterbirds (landbirds) also appear on land (water) backgrounds. As in past work, we use the background as the sensitive feature. Further, we use waterbirds as the positive class and landbirds as the negative class. The dataset is split into training, validation and test sets with 4795, 1199 and 5794 images in each dataset respectively.

We follow the training methodology described in (Sagawa et al., 2020b) to train a deep network for this dataset. First, a fixed pre-trained ResNet-18 model is used to extract a d-dimensional feature vector μ. This feature vector is then converted into an m-dimensional feature $\mu' = ReLU(U\mu)$, where $U \in \mathbb{R}^{m \times d}$ is a random matrix with Gaussian entries. A logistic regression classifier is trained on μ'. Model width is controlled by varying m, the dimensionality of μ', from 10 to $10,000$.



(a) Under-parameterized (b) Over-parameterized

Figure 3: We show the progress of primary loss and FNR_{gap}, evaluated on training dataset, versus SGD steps during MinDiff optimization ($\lambda = 1.5$) for under-parameterized and over-parameterized models on CelebA dataset. We find that over-parameterized models over-fits to the training data and achieve zero FNR_{gap}, thus turning off MinDiff optimization. Whereas, FNR_{gap} is positive in the under-parameterized models, allowing for MinDiff optimization to be effective.

4.2 CelebA Dataset

The CelebA dataset consists of 202,599 celebrity face images annotated with 40 binary attributes including gender, hair colour, hair style, eyeglasses, etc. In our experiments, we set the target label $\mathcal{Y}$ to be hair color, which is either blond ($Y = 1$) or non-blond ($Y = 0$), and the sensitive attribute to be gender. Blond individuals constitute only 15% of this dataset, and only 6% of blond individuals are men. Consequently baseline models make disproportionately large errors on blond men versus blond women. The objective of MinDiff training is to minimize the FNR_{gap} between blond men and blond women [1]. The dataset is split into training, validation and test sets with 162770, 19867 and 19962 images, respectively.

We used the ResNet-preact-18 model for this dataset, and vary model capacity by uniformly scaling the number of channels in all layers.

4.3 Hyper-parameters and Training Details

Waterbirds We train for a total of $30,000$ gradient steps using the Adam optimizer. For our baseline experiments, we set batch size to 128 and use a learning rate schedule with initial learning rate = 0.01 and decay factor of 10 for every $10,000$ gradient steps. We ran every experiment 10 times with random initializations and report the average of all the runs. We trained and evaluated all the models using the Waterbirds dataset on an Intel Xeon Platinum 8268 CPU (24 cores, 2.9 GHz).

[1] Note that the MinDiff paper uses False Positive Rate, but this is totally arbitrary here since the class labels are arbitrary.

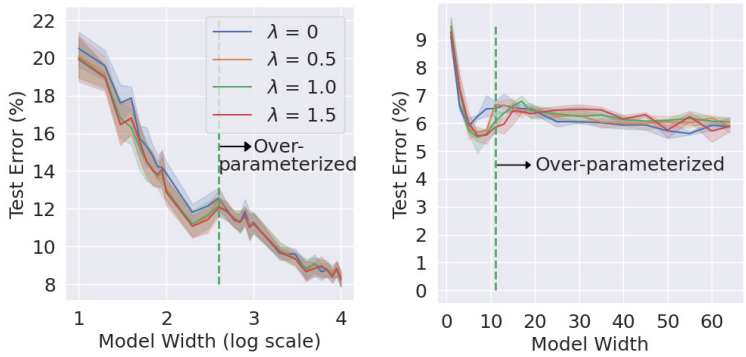
CelebA We train for a total of 48,000 gradient steps using the Adam optimizer. We set the baseline batch size to 128 and adopt a learning rate scheduler with initial learning rate of 0.0001 and decay factor of 10 for every 16,000 gradient steps. In our experiments, we varied the number of channels in the first ResNet-preact-18 block from 1 to 64 (the number of channels is then scaled up by two in each block). We trained all the models using the CelebA dataset on NVIDIA 4 x V100 (32 GB) GPU cards.

For both datasets, we performed MinDiff training with values of $\lambda = \{0.0, 0.5, 1.0, 1.5\}$, where recall that λ controls the importance of the fairness objective. $\lambda = 0.0$ corresponds to training with the primary loss only, and we refer to the resulting model as the baseline model. To study the effect of batch sizing, we trained additional models with batch sizes $\{8, 32\}$. We explored with three different weight decay strengths $= \{0.001, 0.1, 10.0\}$ and two different flood levels, $b \in \{0.05, 0.1\}$. We report the average $\pm$ 95% confidence interval in all figures and tables.

5 Experimental Results

5.1 Identifying the Interpolation Threshold

To distinguish between under- and over-parameterized models, we begin by identifying the interpolation threshold; the point where the model is sufficiently large to interpolate the training data (achieve zero error). Figure 1(a) and Figure 1(b) show the training and test error curves for baseline training versus model width for the Waterbirds and CelebA datasets, respectively, showing interpolation thresholds at model widths of 400 for Waterbirds and 11 for CelebA. On both the datasets, we also observe the double descent phenomenon (Nakkiran et al., 2020), where the test error *decreases* with increasing model capacity beyond the interpolation threshold. That is, for the original model (training without MinDiff), the largest models provide high accuracy.



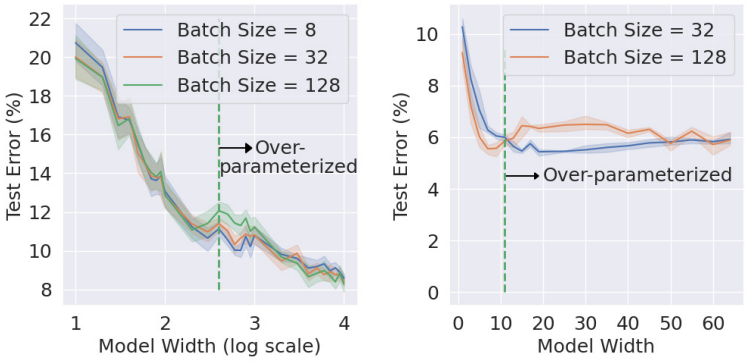
(a) Waterbirds (b) CelebA

Figure 4: We plot the fairness constrained test error with a $\Delta_{\text{FNR}} \leq 10\%$ constraint on the MinDiff trained models with $\lambda = \{0.0, 0.5, 1.0, 1.5\}$ on (a) Waterbirds and (b) CelebA datasets respectively. On both these datasets, we find that MinDiff is only effective for under-parameterized models.

5.2 MinDiff Evaluation

We now re-train our models with MinDiff optimization with $\lambda = \{0.0, 0.5, 1.0, 1.5\}$. Figure 2 shows the test error and FNR_{gap} versus model width for the Waterbirds dataset. In the under-parameterized regime, we observe that, increasing the MinDiff weights significantly reduces the FNR_{gap} with only a small drop in test accuracy. However, in the over-parameterized regime, we find that MinDiff training has no impact on either test error or the FNR_{gap}.

Table 1 shows the test error and FNR_{gap} for selected under- and over-parameterized models on the CelebA dataset trained with MinDiff. Our conclusions are qualitatively the same as for Waterbirds. We find that,



(a) Waterbirds (b) CelebA

Figure 5: We plot the fairness constrained test error with a $\Delta_{\text{FNR}} \leq 10\%$ constraint on the MinDiff trained model with $\lambda = 1.5$ on (a) Waterbirds and (b) CelebA datasets respectively. We find that smaller batch sizes are only effective around the interpolation threshold.

Table 1: We pick three under-parameterized and four over-parameterized model widths and report the average test error and FNR_{gap} for baseline training and several values of λ on CelebA dataset. We find that, in the under-parameterized models, the drop in FNR_{gap} for MinDiff training vs baseline training is more compared to that in the over-parameterized models. We report each data point by averaging over 10 runs.

Width	$\lambda = 0$		$\lambda = 0.5$		$\lambda = 1.0$		$\lambda = 1.5$	
	Error	FNR Gap	Error	FNR Gap	Error	FNR Gap	Error	FNR Gap
5 (Under-)	4.74 ± 0.07	48.77 ± 1.44	5.55 ± 0.39	39.85 ± 3.82	5.46 ± 0.36	40.45 ± 5.58	5.47 ± 0.34	37.06 ± 5.88
7 (Under-)	5.07 ± 0.07	47.35 ± 1.53	5.76 ± 0.5	40.27 ± 2.45	5.64 ± 0.52	42.7 ± 3.12	5.48 ± 0.4	43.05 ± 4.2
9 (Under-)	5.31 ± 0.07	44.78 ± 1.72	5.72 ± 0.4	42.37 ± 1.75	5.77 ± 0.53	40.91 ± 3.28	5.52 ± 0.42	42.43 ± 3.74
13 (Over-)	5.52 ± 0.07	42.79 ± 2.03	5.57 ± 0.1	42.16 ± 2.33	5.66 ± 0.17	42.53 ± 0.98	5.67 ± 0.32	43.07 ± 2.12
19 (Over-)	5.36 ± 0.09	43.98 ± 1.81	5.33 ± 0.05	42.52 ± 1.25	5.42 ± 0.07	42.63 ± 1.01	5.45 ± 0.11	41.34 ± 1.41
55 (Over-)	4.98 ± 0.07	44.14 ± 1.55	5.05 ± 0.07	42.46 ± 1.28	5.24 ± 0.06	42.65 ± 1.85	5.47 ± 0.15	41.93 ± 1.88
64 (Over-)	4.99 ± 0.06	42.54 ± 2.01	5.11 ± 0.06	43.12 ± 2.04	5.28 ± 0.12	41.8 ± 2.32	5.40 ± 0.15	41.56 ± 2.7

MinDiff training significantly reduces the FNR_{gap} compared to the baseline model in the under-parameterized regime, for example, from a 48% FNR gap without MinDiff to a 37% FNR gap with $\lambda = 1.5$ for a model width of 5. On the other hand, MinDiff training compared with the baseline model has a negligible effect on both test error and FNR_{gap} in the over-parameterized regime, sometimes even resulting in a (small) increase in both.

Table 2: We tabulate the fairness constrained test error (with a $\Delta_{\text{FNR}} \leq 10\%$ constraint) of early stopped MinDiff models (trained with $\lambda = \{0.5, 1.0, 1.5\}$) for different widths. We compare two methods for early stopping (es) based on the stopping criterion: primary validation loss ($\text{es}(\mathcal{L}_P)$) and total validation loss ($\text{es}(\mathcal{L}_T)$). We find that, for CelebA dataset, MinDiff + $\text{es}(\mathcal{L}_P)$ is better than MinDiff + $\text{es}(\mathcal{L}_T)$.

	$\lambda = 0.5$			$\lambda = 1.0$			$\lambda = 1.5$		
	Width = 5	Width = 19	Width = 64	Width = 5	Width = 19	Width = 64	Width = 5	Width = 19	Width = 64
No es	6.36 ± 0.51	6.54 ± 0.12	5.88 ± 0.13	6.25 ± 0.43	6.43 ± 0.12	6.06 ± 0.06	6.36 ± 0.35	6.54 ± 0.18	6.07 ± 0.2
$\text{es}(\mathcal{L}_P)$	6.17 ± 0.34	5.91 ± 0.26	5.45 ± 0.22	6.32 ± 0.28	5.82 ± 0.32	4.79 ± 0.17	6.37 ± 0.25	5.63 ± 0.3	4.70 ± 0.07
$\text{es}(\mathcal{L}_T)$	7.76 ± 0.56	6.47 ± 0.31	5.76 ± 0.61	7.79 ± 0.47	7.18 ± 0.58	6.05 ± 0.53	8.65 ± 0.65	7.38 ± 0.77	6.80 ± 0.64

We observe that MinDiff performs poorly for over-parameterized models: Figure 3 shows how the primary loss and FNR_{gap}'s change during SGD steps for under-parameterized and over-parameterized models on the CelebA dataset. We note that at the beginning of training, the FNR_{gap} is small because randomly initialized models make random predictions. These random predictions are fair w.r.t the error rates but not necessarily fair according to other criteria like Demographic Parity. As training progresses, the over-parameterized model eventually over-fits the training data at around 20,000 steps, achieving zero primary loss. This model also appears to be trivially fair from the standpoint of the training data—observe that at the same point, the

FNR_{gap} also goes to zero, and no further optimization takes place. On the other hand, the FNR_{gap} during training remains positive for the under-parameterized model.

Figure 4 plots the fairness constrained test error with a $\Delta_{\text{FNR}} \leq 10\%$ constraint using post-training threshold correction on the MinDiff trained models. We can again observe that MinDiff is only effective for under-parameterized models. For CelebA, the lowest fairness constrained test error is actually achieved by an under-parameterized model. In other words, *achieving fairness via MinDiff optimization requires careful selection of model width, including exploring the under-parameterized regime.*

Table 3: We tabulate the fairness constrained test error (with a $\Delta_{\text{FNR}} \leq 10\%$ constraint) for different regularization schemes used in conjunction with MinDiff optimization ($\lambda = 1.5$) on CelebA dataset. The performance of best regularizer is highlighted in bold for each model width. Notation: wd is weight decay, es($\mathcal{L}_P$) is early stopping w.r.t primary loss and fl is flooding

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
$\lambda = 0$	5.92 ± 0.19	6.51 ± 0.11	5.76 ± 0.1
$\lambda = 1.5$	6.36 ± 0.35	6.54 ± 0.18	6.07 ± 0.2
$\lambda = 1.5$ + wd = 0.001	$\mathbf{5.82 \pm 0.2}$	6.53 ± 0.16	5.08 ± 0.15
$\lambda = 1.5$ + wd = 0.1	6.15 ± 0.15	5.82 ± 0.16	6.57 ± 0.1
$\lambda = 1.5$ + es($\mathcal{L}_P$)	6.37 ± 0.25	5.63 ± 0.3	4.70 ± 0.07
$\lambda = 1.5$ + fl = 0.05	6.28 ± 0.25	6.30 ± 0.23	5.49 ± 0.17
$\lambda = 1.5$ + fl = 0.1	6.30 ± 0.3	$\mathbf{5.25 \pm 0.25}$	$\mathbf{4.67 \pm 0.12}$

5.3 Impact of Regularization in Over-parameterized Regime

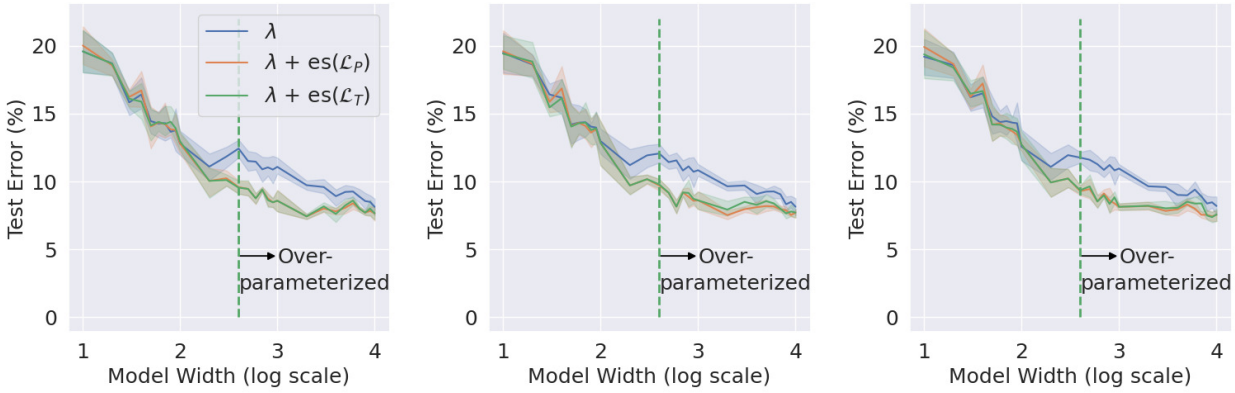
We now examine if additional regularization can help improve the fairness of MinDiff-regularized models in the over-parameterized regime. Unless otherwise stated, in all subsequent evaluations we perform post-training threshold correction with a $\Delta_{\text{FNR}} \leq 10\%$ constraint and compare fairness constrained test error.

Batch sizing only helps around the interpolation threshold. Small batch sizes cause primary training loss curves to converge more slowly, potentially providing more opportunity for MinDiff optimizations. In Figure 5(a) and Figure 5(b), we plot fairness constrained test error curves versus model widths for different batch sizes on the Waterbirds and CelebA datasets, respectively. We find that smaller batch sizes improve fairness constrained test error only around the interpolation threshold for both the datasets, but do not noticeably benefit smaller and larger models. On further examination, we note the benefits around the interpolation threshold are because smaller batch sizes induce stronger regularization effects and push the interpolation threshold to the right (see Appendix Figure 11). As a result, MinDiff is effective on a slightly increased range of model widths. However, other than this behaviour, we see no other benefits of using batch sizing as a regularizer and do not explore it further.

Early stopping criterion. We evaluate two methods for early stopping. The first uses the primary validation loss (MinDiff+es($\mathcal{L}_P$)) as a stopping criterion, while the second uses total validation loss for stopping (MinDiff+es($\mathcal{L}_T$)). Figure 6 plots fairness constrained test error versus model width for these two schemes and different values of λ on Waterbirds, and Table 2 shows the same data for CelebA. We find that both schemes improve fairness for over-parameterized models, but have limited impact in the under-parameterized regime. For Waterbirds, the differences between the two are small, although using primary validation loss as the stopping criterion (MinDiff+es($\mathcal{L}_P$)) is marginally better than using total validation loss (MinDiff+es($\mathcal{L}_T$)). However, for CelebA, we find that primary loss stopping criterion (MinDiff+es($\mathcal{L}_P$)) is substantially better than total loss stopping criterion (MinDiff+es($\mathcal{L}_T$)), especially for large models. Thus, we use the former for the remainder of our experiments.

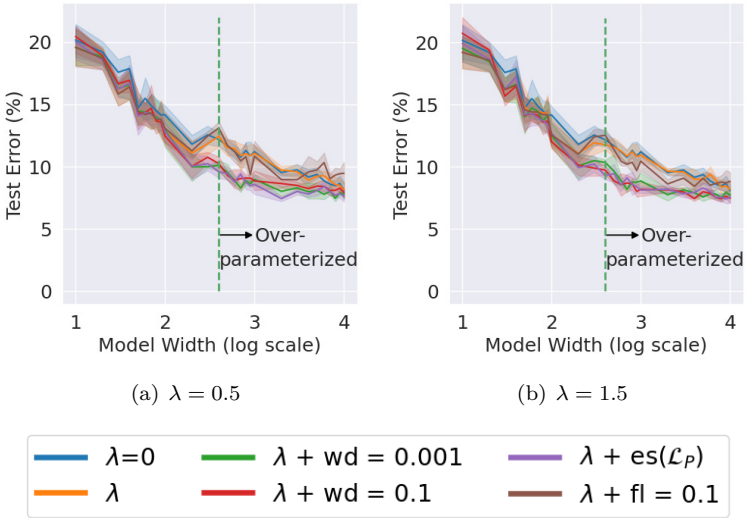
Comparing regularization methods on Waterbirds. In Figure 7, we plot the fairness constrained test error versus model width for different regularization schemes including early stopping (λ+es), weight decay with two different values (λ+wd=0.001, λ+wd=0.1) and flooding (λ+fl) on Waterbirds. We find that the early

stopping and weight decay regularizes substantially improve fairness for models just below the interpolation threshold and all over-parameterized models. In contrast, flooding shows only small improvements in fairness. For $\lambda = 0.5$, we find that early stopping is the best across the board. For $\lambda = 1.5$, either early stopping and weight decay are the best depending on model width, although the differences are small.



(a) $\lambda = 0.5$ (b) $\lambda = 1.0$ (c) $\lambda = 1.5$

Figure 6: We plot the fairness constrained test error (with a $\Delta_{\text{FNR}} \leq 10\%$ constraint) of early stopped MinDiff models (trained with $\lambda = \{0.5, 1.0, 1.5\}$) for several model widths. We compare two methods for early stopping (es) based on the stopping criterion: primary validation loss (es($\mathcal{L}_P$)) and total validation loss (es($\mathcal{L}_T$)). We find that, for Waterbirds dataset, using either stopping criterion will significantly improve fairness constrained error, especially in the over-parameterized regime.



(a) $\lambda = 0.5$ (b) $\lambda = 1.5$

Figure 7: We plot fairness constrained test error (with a $\Delta_{\text{FNR}} \leq 10\%$ constraint) versus model widths for different regularization schemes used in conjunction with MinDiff optimization on Waterbirds dataset. We find that early stopping and weight decay are preferred choice of regularizers for Waterbirds dataset. Notation: wd is weight decay, es($\mathcal{L}_P$) is early stopping w.r.t primary loss and fl is flooding.

Comparing regularization methods on CelebA. Table 3 compares different regularization schemes on the CelebA dataset for three model widths and for $\lambda = 0.5$, $\lambda = 1.0$ and $\lambda = 1.5$, respectively. For the smallest model, we find that weight decay schemes result in the lowest fairness constrained error, while for the large over-parameterized model, flooding works best. Comparing across model widths, we find that for each λ value, the largest model with flooding provides the overall lowest fairness constrained test error. Recalling that flooding was ineffective on Waterbirds, we conclude that no one regularizer works best across datasets

and model widths, but additional regularization in general can restore the benefits of over-parameterization with fairness constraints.

6 Conclusion

In this paper, we have critically examined the performance MinDiff, an in-training fairness regularization technique implemented within TensorFlow's Responsible AI toolkit, with respect to DNN model complexity, with a particular eye towards over-parameterized models. On two datasets commonly used in fairness evaluations, we find that although MinDiff improves the fairness of under-parameterized models relative to baseline, it is ineffective in improving fairness for over-parameterized models. As a result, we find that for one of our two datasets, under-parameterized MinDiff models have lower fairness constrained test error than their over-parameterized counterparts, suggesting that time-consuming searches for best model size might be necessary when MinDiff is used with the goal of training a fair model.

To address these concerns, we explore traditional batch sizing, weight decay and early stopping regularizers to improve MinDiff training, in addition to flooding, a recently proposed method that is evaluated for the first time in the context of fair training. We find that batch sizing is ineffective in improving fairness except for model widths near the interpolation threshold. The other regularizers do improve fairness for over-parameterized models, but the best regularizer depends on the dataset and model size. In particular, flooding results in the fairest models on the CelebA dataset, suggesting its utility in the fairness toolkit. Finally, we show that with appropriate choice of regularizer, over-parameterized models regain their benefits over under-parameterized counterparts even from a fairness lens.

Availability

Code with README.txt file is available at: `https://anonymous.4open.science/r/fairml-EBEE/`

References

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification, 2018.

TensorFlow Responsible AI. Responsible ai toolkit: Tensorflow. URL https://www.tensorflow.org/responsible_ai.

Masashi Sode Akihiko Fukuchi, Yoko Yabe. Fairtorch. https://github.com/wbawakate/fairtorch, 2020.

David Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=3q5IqUrkcF.

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. Data decisions and theoretical implications when adversarially learning fair representations, 2017.

Alex Beutel, Jilin Chen, Tulsee Doshi, Hai Qian, Allison Woodruff, Christine Luu, Pierre Kreitmann, Jonathan Bischof, and Ed H. Chi. Putting fairness principles into practice: Challenges, metrics, and improvements. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pp. 453–459, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450363242. doi: 10.1145/3306618.3314234. URL https://doi.org/10.1145/3306618.3314234.

Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In Sorelle A. Friedler and Christo Wilson (eds.), *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pp. 77–91. PMLR, 23–24 Feb 2018. URL https://proceedings.mlr.press/v81/buolamwini18a.html.

Valeriia Cherepanova, Vedant Nanda, Micah Goldblum, John P. Dickerson, and Tom Goldstein. Technical challenges for training fair neural networks, 2021.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. doi: 10.1089/big.2016.0047. URL https://doi.org/10.1089/big.2016.0047. PMID: 28632438.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, Luca Oneto, and Massimiliano Pontil. *Leveraging Labeled and Unlabeled Data for Consistent Fair Binary Classification.* Curran Associates Inc., Red Hook, NY, USA, 2019.

Guillem Collell, Drazen Prelec, and Kaustubh R. Patil. Reviving threshold-moving: a simple plug-in bagging ensemble for binary and multiclass imbalanced data. *ArXiv*, abs/1606.08698, 2016.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '18, pp. 67–73, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360128. doi: 10.1145/3278721.3278729. URL https://doi.org/10.1145/3278721.3278729.

Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 2796–2806, Red Hook, NY, USA, 2018. Curran Associates Inc.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pp. 259–268, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336642. doi: 10.1145/2783258.2783311. URL https://doi.org/10.1145/2783258.2783311.

Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11662.

Patrick Grother, George Quinn, and P Phillips. Report on the evaluation of 2d still-image face recognition algorithms, 2010-06-17 2010.

Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016a. URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016b. URL https://proceedings.neurips.cc/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

Tatsunori B. Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *ICML*, 2018.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4604–4614. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/ishida20a.html.

Ray Jiang, Aldo Pacchiano, Tom Stepleton, Heinrich Jiang, and Silvia Chiappa. Wasserstein fair classification. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 862–872. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/jiang20a.html.

Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1548–1558, 2021.

Amir E. Khandani, Adlar J. Kim, and Andrew W. Lo. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance*, 34(11):2767–2787, 2010. ISSN 0378-4266. doi: https://doi.org/10.1016/j.jbankfin.2010.06.001. URL https://www.sciencedirect.com/science/article/pii/S0378426610002372.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. In J. Moody, S. Hanson, and R. P. Lippmann (eds.), *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf.

Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning, 2020.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

David Madras, Elliot Creager, Toniann Pitassi, and Richard Zemel. Learning adversarially fair and transferable representations, 2018.

Subha Maity, Saptarshi Roy, Songkai Xue, Mikhail Yurochkin, and Yuekai Sun. How does overparametrization affect performance on minority groups?, 2022. URL https://arxiv.org/abs/2206.03515.

Natalia Martinez, Martin Bertran, and Guillermo Sapiro. Minimax pareto fairness: A multi objective perspective, 2020.

Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=37nvvqkCo5.

Aditya Krishna Menon, Ankit Singh Rawat, and Sanjiv Kumar. Overparameterisation and worst-case generalisation: friend or foe? In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=jphnJNOwe36.

N. Morgan and H. Bourlard. *Generalization and Parameter Estimation in Feedforward Nets: Some Experiments*, pp. 630–637. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1558601007.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1g5sA4twr.

Mei Ngan and Patrick Grother. Face recognition vendor test (frvt) - performance of automated gender classification algorithms, 2015-04-20 2015.

Manisha Padala and Sujit Gujar. Fnnc: Achieving fairness through neural networks. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, IJCAI'20, 2021. ISBN 9780999241165.

Alan Pham, Eunice Chan, Vikranth Srivatsa, Dhruba Ghosh, Yaoqing Yang, Yaodong Yu, Ruiqi Zhong, Joseph E. Gonzalez, and Jacob Steinhardt. The effect of model size on worst-group generalization, 2021. URL https://arxiv.org/abs/2112.04094.

Flavien Prost, Hai Qian, Qiuwen Chen, Ed H. Chi, Jilin Chen, and Alex Beutel. Toward a better trade-off between performance and fairness with kernel-based distribution matching, 2019.

Novi Quadrianto, Viktoriia Sharmanska, and Oliver Thomas. Discovering fair representations in the data domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Hee Jung Ryu, Hartwig Adam, and Margaret Mitchell. Inclusivefacenet: Improving face attribute detection with race and gender diversity, 2018.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=ryxGuJrFvS.

Shiori Sagawa, Aditi Raghunathan, Pang Wei Koh, and Percy Liang. An investigation of why overparameterization exacerbates spurious correlations. In *ICML*, pp. 8346–8356, 2020b. URL http://proceedings.mlr.press/v119/sagawa20a.html.

Yash Savani, Colin White, and Naveen Sundar Govindarajulu. Intra-processing methods for debiasing neural networks, 2020.

Candice Schumann, Jeffrey S. Foster, Nicholas Mattei, and John P. Dickerson. We need fairness and explainability in algorithmic hiring. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pp. 1716–1720, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

Ashudeep Singh and Thorsten Joachims. Fairness of exposure in rankings. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &; Data Mining*, KDD '18, pp. 2219–2228, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220088. URL https://doi.org/10.1145/3219819.3220088.

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rq_Qr0c1Hyo.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Christina Wadsworth, Francesca Vera, and Chris Piech. Achieving fairness through adversarial learning: an application to recidivism prediction, 2018.

Yoav Wald, Gal Yona, Uri Shalit, and Yair Carmon. Malign overfitting: Interpolation and invariance are fundamentally at odds. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022. URL https://openreview.net/forum?id=1xadmcm2CC.

Mei Wang and Weihong Deng. Mitigate bias in face recognition using skewness-aware reinforcement learning, 2019.

Zeyu Wang, Klint Qinami, Ioannis Christos Karakozis, Kyle Genova, Prem Nair, Kenji Hata, and Olga Russakovsky. Towards fairness in visual recognition: Effective strategies for bias mitigation, 2020.

Dennis Wei, Karthikeyan Natesan Ramamurthy, and Flavio Calmon. Optimized score transformation for fair classification. In Silvia Chiappa and Roberto Calandra (eds.), *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 1673–1683. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/wei20a.html.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019. URL http://jmlr.org/papers/v20/18-262.html.

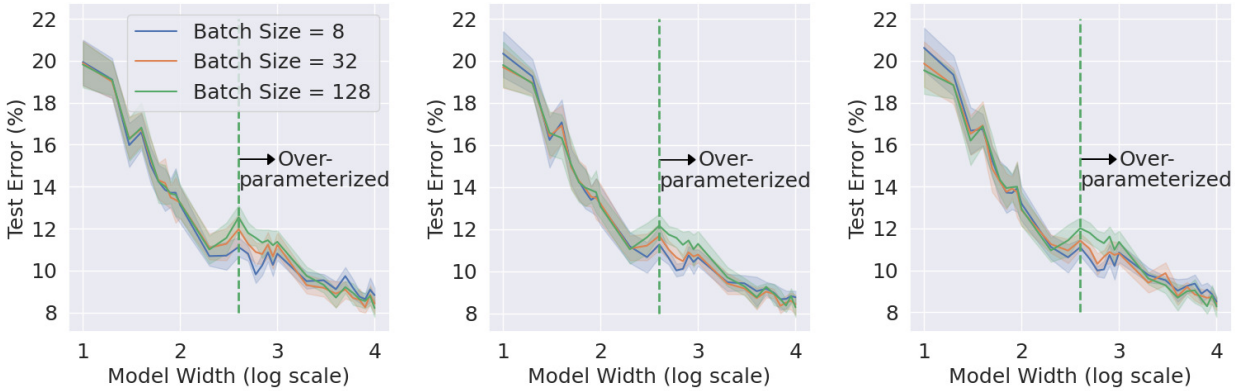
Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning, 2018.

Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2006. doi: 10.1109/TKDE.2006.17.

A Impact of Regularization in Over-parameterized Regime

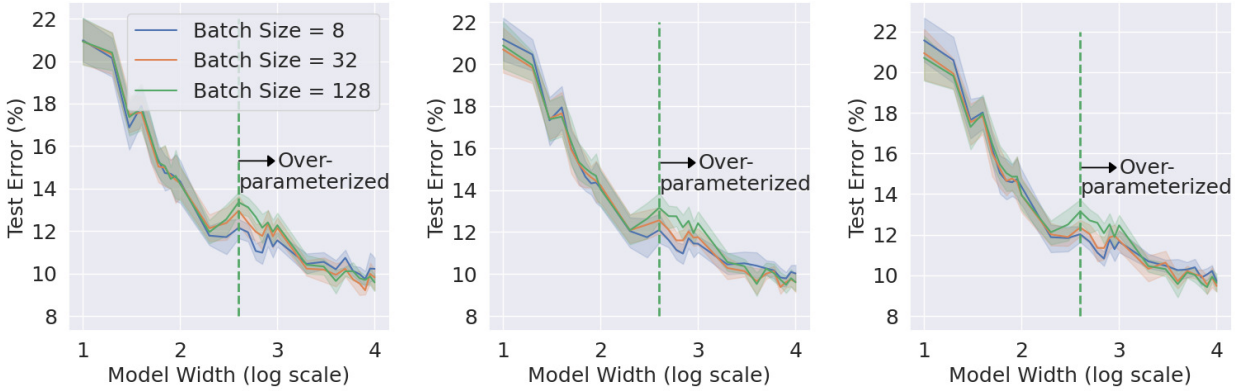
A.1 Batch Size

A.1.1 Waterbirds



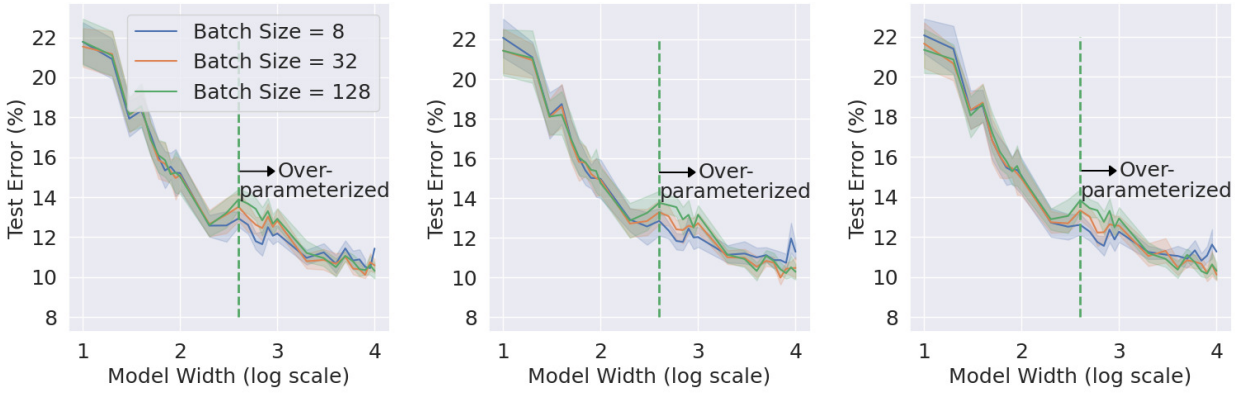
(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 8: Batch size - THR with fairness constraint $< 10\%$



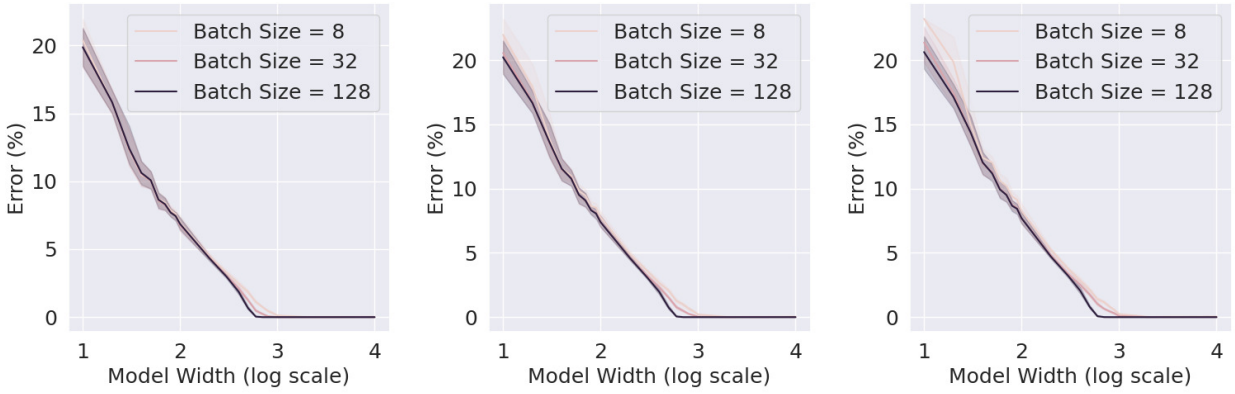
(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 9: Batch size - THR with fairness constraint $< 5\%$



(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 10: Batch size - THR with fairness constraint $< 1\%$

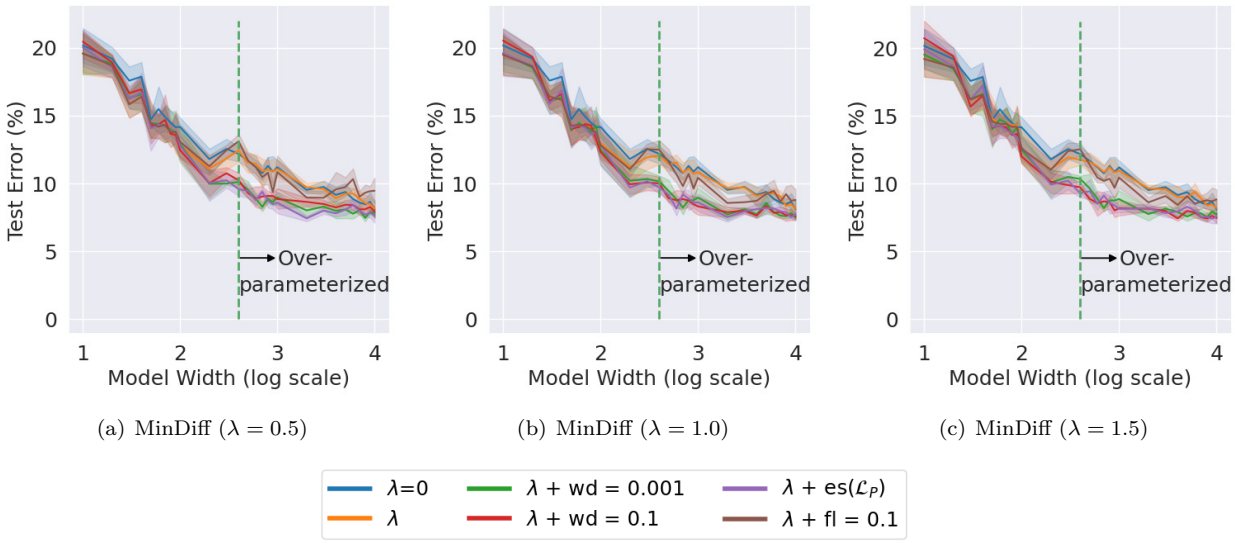


(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 11: Training loss vs model widths for different batch sizes

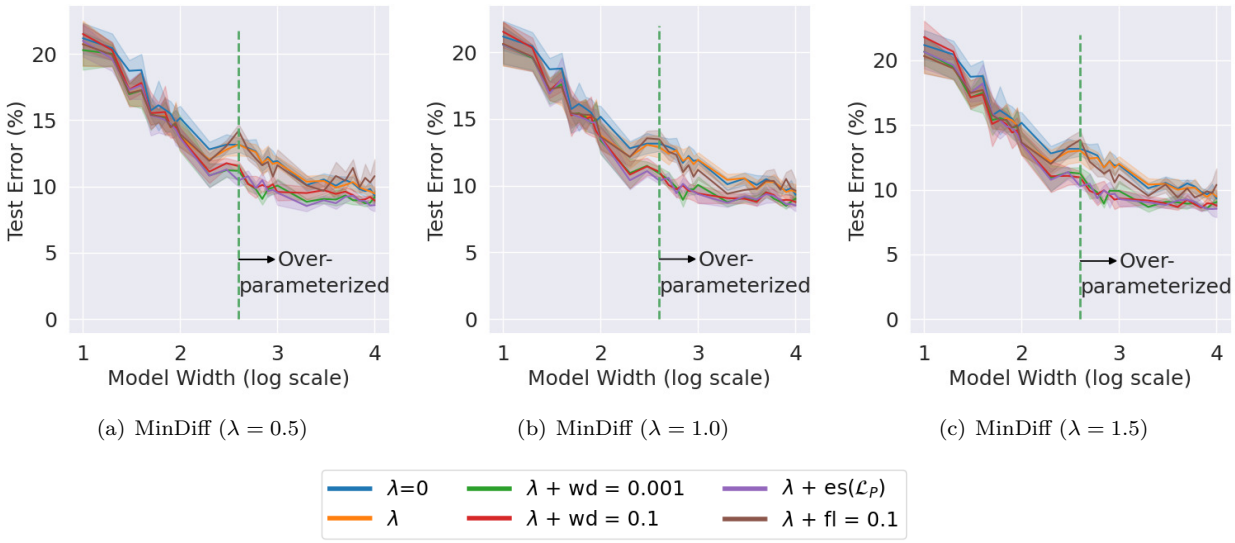
A.2 All Other Regularizers

A.2.1 Waterbirds



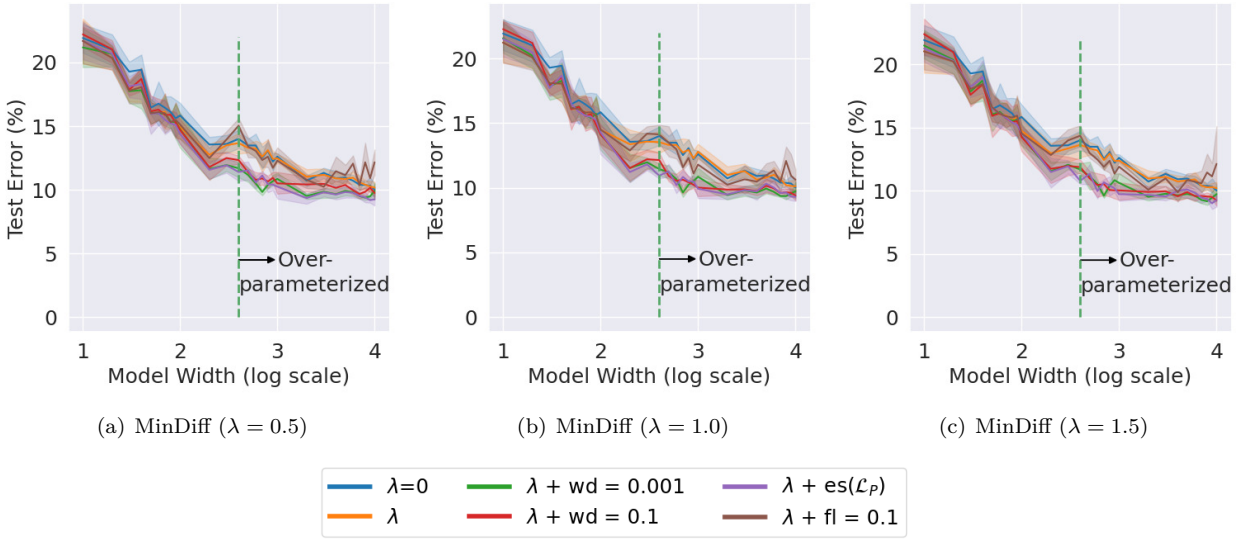
(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 12: Regularization - THR with fairness constraint $< 10\%$



(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 13: Regularization - THR with fairness constraint $< 5\%$



(a) MinDiff ($\lambda = 0.5$) (b) MinDiff ($\lambda = 1.0$) (c) MinDiff ($\lambda = 1.5$)

Figure 14: Regularization - THR with fairness constraint $< 1\%$

A.2.2 CelebA

Table 4: We tabulate the fairness constrained test error (with a $\Delta_{FNR} \leq 10\%$ constraint) for different regularization schemes used in conjunction with MinDiff optimization ($\lambda = 0.5$) on CelebA dataset. The performance of best regularizer is highlighted in bold for each model width. Notation: wd is weight decay, es($\mathcal{L}_P$) is early stopping w.r.t primary loss and fl is flooding

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
$\lambda = 0$	5.92 ± 0.27	6.51 ± 0.16	5.76 ± 0.14
$\lambda = 0.5$	6.36 ± 0.72	6.54 ± 0.17	5.88 ± 0.19
$\lambda = 0.5$ + wd $= 0.001$	5.89 ± 0.32	6.94 ± 0.18	6.25 ± 0.48
$\lambda = 0.5$ + wd $= 0.1$	$\mathbf{5.66 \pm 0.18}$	$\mathbf{5.27 \pm 0.16}$	6.40 ± 0.18
$\lambda = 0.5$ + es($\mathcal{L}_P$)	6.17 ± 0.48	5.91 ± 0.37	5.45 ± 0.32
$\lambda = 0.5$ + fl $= 0.05$	6.45 ± 0.69	6.53 ± 0.21	6.16 ± 0.09
$\lambda = 0.5$+ fl $= 0.1$	6.34 ± 0.63	5.75 ± 0.35	$\mathbf{4.80 \pm 0.28}$

Table 5: We tabulate the fairness constrained test error (with a $\Delta_{FNR} \leq 10\%$ constraint) for different regularization schemes used in conjunction with MinDiff optimization ($\lambda = 1.0$) on CelebA dataset. The performance of best regularizer is highlighted in bold for each model width. Notation: wd is weight decay, es($\mathcal{L}_P$) is early stopping w.r.t primary loss and fl is flooding

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
$\lambda = 0$	5.92 ± 0.27	6.51 ± 0.16	5.76 ± 0.14
$\lambda = 1.0$	6.25 ± 0.61	6.43 ± 0.17	6.06 ± 0.09
$\lambda = 1.0$ + wd = 0.001	5.94 ± 0.39	6.75 ± 0.35	5.90 ± 0.24
$\lambda = 1.0$ + wd = 0.1	$\mathbf{5.80 \pm 0.17}$	5.55 ± 0.19	6.42 ± 0.12
$\lambda = 1.0$ + es($\mathcal{L}_P$)	6.32 ± 0.40	5.82 ± 0.46	4.79 ± 0.24
$\lambda = 1.0$ + fl = 0.05	6.30 ± 0.68	6.43 ± 0.17	5.87 ± 0.11
$\lambda = 1.0$ + fl = 0.1	6.33 ± 0.52	$\mathbf{5.4 \pm 0.35}$	$\mathbf{4.77 \pm 0.13}$

Table 6: CelebA (MinDiff: 0.5, FNR Gap $\leq 5\%$, Batch Size: 128)

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
Original Model	6.22 ± 0.31	6.81 ± 0.21	6.10 ± 0.14
MinDiff	6.72 ± 0.70	6.78 ± 0.16	6.18 ± 0.25
MinDiff + wd = 0.001	6.20 ± 0.35	7.29 ± 0.21	6.51 ± 0.55
MinDiff + wd = 0.1	$\mathbf{5.84 \pm 0.13}$	$\mathbf{5.51 \pm 0.15}$	6.70 ± 0.14
MinDiff + es	6.48 ± 0.48	6.14 ± 0.38	5.61 ± 0.30
MinDiff + fl = 0.05	6.73 ± 0.70	6.80 ± 0.21	6.47 ± 0.11
MinDiff + fl = 0.1	6.65 ± 0.67	5.99 ± 0.37	$\mathbf{5.03 \pm 0.29}$

Table 7: CelebA (MinDiff: 1.0, FNR Gap $\leq 5\%$, Batch Size: 128)

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
Original Model	6.22 ± 0.31	6.81 ± 0.21	6.10 ± 0.14
MinDiff	6.57 ± 0.56	6.74 ± 0.20	6.35 ± 0.07
MinDiff + wd = 0.001	6.26 ± 0.45	7.11 ± 0.34	6.21 ± 0.23
MinDiff + wd = 0.1	$\mathbf{6.15 \pm 0.18}$	5.82 ± 0.14	6.70 ± 0.14
MinDiff + es	6.60 ± 0.40	6.15 ± 0.50	5.05 ± 0.18
MinDiff + fl = 0.05	6.57 ± 0.70	6.74 ± 0.14	6.15 ± 0.15
MinDiff + fl = 0.1	6.58 ± 0.43	$\mathbf{5.67 \pm 0.38}$	$\mathbf{4.92 \pm 0.06}$

Table 8: CelebA (MinDiff: 1.5, FNR Gap $\leq 5\%$, Batch Size: 128)

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
Original Model	6.22 ± 0.31	6.81 ± 0.21	6.10 ± 0.14
MinDiff	6.65 ± 0.46	6.90 ± 0.30	6.40 ± 0.26
MinDiff + wd = 0.001	$\mathbf{6.15 \pm 0.41}$	6.85 ± 0.30	5.33 ± 0.21
MinDiff + wd = 0.1	6.45 ± 0.18	6.11 ± 0.13	6.81 ± 0.10
MinDiff + es	6.73 ± 0.43	5.89 ± 0.45	4.88 ± 0.05
MinDiff + fl = 0.05	6.58 ± 0.39	6.64 ± 0.40	5.77 ± 0.18
MinDiff + fl = 0.1	6.68 ± 0.44	$\mathbf{5.56 \pm 0.41}$	$\mathbf{4.86 \pm 0.14}$

Table 9: CelebA (MinDiff: 0.5, FNR Gap $\leq 1\%$, Batch Size: 128)

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
Original Model	6.57 ± 0.32	7.13 ± 0.22	6.31 ± 0.12
MinDiff	6.96 ± 0.72	7.07 ± 0.14	6.50 ± 0.26
MinDiff + wd = 0.001	6.52 ± 0.35	7.57 ± 0.24	6.86 ± 0.55
MinDiff + wd = 0.1	**6.12 ± 0.13**	**5.82 ± 0.18**	6.94 ± 0.20
MinDiff + es	6.71 ± .048	6.41 ± 0.42	5.85 ± 0.29
MinDiff + fl = 0.05	7.04 ± 0.70	7.09 ± 0.24	6.78 ± 0.10
MinDiff + fl = 0.1	6.94 ± 0.67	6.20 ± 0.42	**5.17 ± 0.28**

Table 10: CelebA (MinDiff: 1.0, FNR Gap $\leq 1\%$, Batch Size: 128)

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
Original Model	6.57 ± 0.32	7.13 ± 0.22	6.31 ± 0.12
MinDiff	6.76 ± 0.54	7.00 ± 0.18	6.55 ± 0.14
MinDiff + wd = 0.001	6.54 ± 0.46	7.43 ± 0.41	6.46 ± 0.20
MinDiff + wd = 0.1	**6.43 ± 0.22**	6.11 ± 0.12	6.97 ± 0.17
MinDiff + es	6.86 ± 0.42	6.40 ± 0.53	5.24 ± 0.21
MinDiff + fl = 0.05	6.87 ± 0.66	7.04 ± 0.17	6.33 ± 0.15
MinDiff + fl = 0.1	6.90 ± 0.41	**5.94 ± 0.33**	**5.09 ± 0.08**

Table 11: CelebA (MinDiff: 1.5, FNR Gap $\leq 1\%$, Batch Size: 128)

Method	Width = 5	Width = 19	Width = 64
	Under-	Over-	Over-
Original Model	6.57 ± 0.32	7.13 ± 0.22	6.31 ± 0.12
MinDiff	6.86 ± 0.49	7.24 ± 0.31	6.64 ± 0.26
MinDiff + wd = 0.001	**6.42 ± 0.45**	7.13 ± 0.30	5.68 ± 0.25
MinDiff + wd = 0.1	6.75 ± 0.22	6.29 ± 0.17	7.05 ± 0.14
MinDiff + es	6.97 ± 0.38	6.12 ± 0.49	5.07 ± 0.06
MinDiff + fl = 0.05	6.85 ± 0.39	6.92 ± 0.42	5.93 ± 0.24
MinDiff + fl = 0.1	6.89 ± 0.45	**5.84 ± 0.42**	**5.04 ± 0.16**